WJSehnert_STT223_FinalCapstone

Jake Sehnert

5/2/2019

Question 1

The joint posterior distribution was derived during the in-class portion of the Final Capstone.

Question 2

The fully conditional posterior distributions for each parameter were derived during the in-class portion of the Final Capstone.

Question 3

In order to draw 10,000 samples from these fully conditional posterior distributions derived in part **1.**, I need to define the Gibbs function sampling function. However, since one of the hyperparameters, α, has an unnormalized fully conditional posterior distribution, I first need to implement the Metropolis-Hastings algorithm to draw samples.

```r
# Define the unnormalized posterior of alpha
alpha.posterior <- function(a, b, thetas) { # Data has thetas in column 1,
betas in column 2
  beta <- b
  dens <- exp(-a)*prod(((thetas^(a-1))*(b^a))/gamma(a))
  return(dens)
}

# Import library to sample from truncated normal distribution
library(truncnorm)

# Proposal distribution
prop.dist.alpha <- function(a, prop.var) {
  rtruncnorm(1, mean=a, sd=sqrt(prop.var), a=0)
}

# Density of proposal
prop.dist.alpha.dens <- function(a, a.mean, prop.var) {
  dtruncnorm(a, mean = a.mean, sd=sqrt(prop.var), a=0)
}

# Metropolis-Hastings Algorithm
metrop <- function(param, thetas, b, alpha.posterior, prop.dist.alpha,
prop.dist.alpha.dens, prop.var, n.iter) {
  # Store sampled alpha values
  alphas <- c()

  # Initialize model
  param.t <- param

  for(t in 1:n.iter) {

    # Draw proposed value of alpha
    param.new <- prop.dist.alpha(param.t, prop.var)

    # Calculate acceptance probability
    u <- runif(1, 0, 1)
    prob.accept <- min(1, (alpha.posterior(param.new, b,
thetas)*(prop.dist.alpha.dens(param.t, param.new,
prop.var)))/(alpha.posterior(param.t, b,
thetas)*(prop.dist.alpha.dens(param.new, param.t, prop.var))))

    if(u < prob.accept) {
      value <- param.new
    } else {
      value <- param.t
    }

    # Store sampled alpha value
    alphas <- c(alphas, value)

    # Update value for next iteration
    param.t <- value
  }

  # Modification for MH-within-Gibbs sampling --> if only drawing one sample,
return the sampled value.
  # If drawing multiple samples, return the list of all samples
  if (length(alphas) == 1) {
    return(alphas[1])
  } else {
    return(alphas)
  }
}
```

Now that I’ve implemented the Metropolis-Hastings algorithm to generate samples for α, I need to define the Gibbs sampling function to draw samples oof the other posteriors.

```
# Define Gibbs sampling function
gibbs <- function(initial, y, t, n.iter) {
  # Initialize variables
  J <- length(y)
```

```r
  l <- length(initial)
  results <- matrix(NA, n.iter, l)
  results[1,] <- initial

  for(i in 2:n.iter) {
    thetas <- results[i-1, 1:7] # Stores all 7 theta_j values
    a <- results[i-1,8]
    b <- results[i-1,9]

    # Draw theta_j samples
    for(j in 1:J) {

        # Find alpha, beta parameters for theta_j's gamma posterior
distribution
        alpha.theta <- y[j] + a
        beta.theta <- t[j] + b

        # Store singular theta_j sample using parameters calculated above
        results[i,j] <- rgamma(1, alpha.theta, beta.theta)
    }

    # Find alpha, beta parameters for beta's gamma posterior distribution
using theta sample
    alpha.beta <- J*a + 0.1
    beta.beta <- 1 + sum(results[i, 1:7])

    # Store singular beta sample from its gamma posterior distribution
    results[i, 9] <- rgamma(1, alpha.beta, beta.beta)

    # Use Metropolis-Hastings algorithm from above to draw singular alpha
sample
    results[i, 8] <- metrop(a, results[i, 1:7], results[i, 9],
alpha.posterior, prop.dist.alpha, prop.dist.alpha.dens, 4, 1)
  }
  return(results)
}
```

No that I’ve defined the Gibbs sampling method, I can proceed with drawing 10,000 samples. However, before sampling, I first need to install the MCMCpack library and store prior information. After doing this, I can proceed with sampling.

```r
## Import library for computation
library(MCMCpack)

## Loading required package: coda

## Loading required package: MASS

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
## ## Copyright (C) 2003-2019 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##

set.seed(5000) # Set seed for reproducability

# Store prior data
yhat <- c(78, 64, 90, 78, 83, 82, 89)
t <- c(14.1, 13.2, 15.4, 14.9, 15.6, 15.2, 16.6)

# Initialize first set of sampling values
initial_1 <- c(rep(.1, 7), 1, 1)

# Draw samples
sample_1 <- gibbs(initial_1, yhat, t, 10000)
```

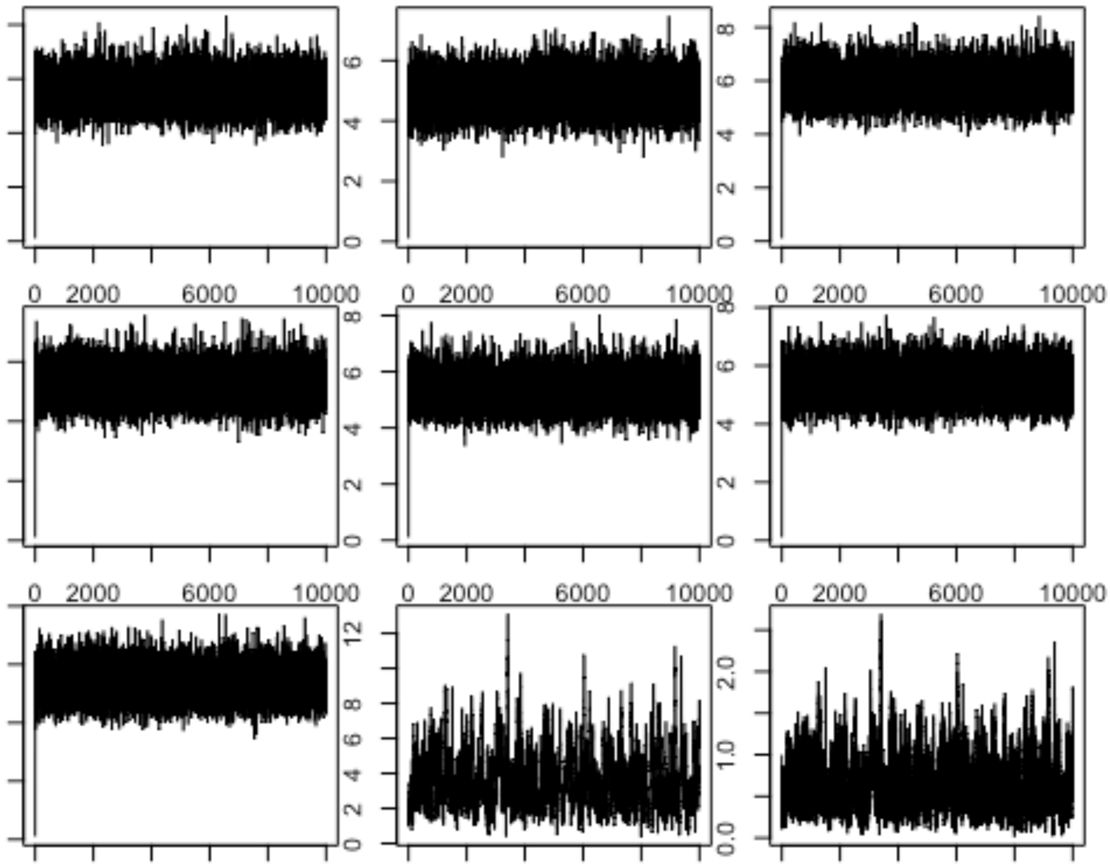
Now that I've sampled each parameter's fully conditinal posterior distribution, I show the posterior means in the table below.

```
means <- matrix(NA, 9, 1)
for (i in 1:9) {
  means[i,1] <- mean(sample_1[,i])
}
rownames(means) <- c("Theta1", "Theta2", "Theta3", "Theta4",
"Theta5","Theta6","Theta7", "Alpha", "Beta")
colnames(means) <- c("Posterior Mean")
means <- as.table(means)
means

##        Posterior Mean
## Theta1      5.5327379
## Theta2      4.8806227
## Theta3      5.8284910
## Theta4      5.2315078
## Theta5      5.3255287
## Theta6      5.4011312
## Theta7      5.3639737
## Alpha       3.4411400
## Beta        0.6263803
```

Question 4

Now that I've drawn 10,000 samples from each parameter's fully conditional posterior distribution, I need to examine whether or not convergence is met. I first do this by constructing a traceplot for each parameter.

```
par(mfrow = c(3, 3))
par(mar = c(1, 1, 1, 1))

for (i in 1:9) {
  plot(sample_1[,i], type="l")
}
```



The traceplots above lead me to believe convergence has been met for each parameter. Each traceplot plots the values of each parameter over their respective iteration where they are calculated. Thus, keeping in mind the properties of an ideal MH algorithm (irreducable, positive recurrent and aperiodic), we want the traceplots to look like a "hairy caterpillar", indicating the sampled values move across the parameter's range over the number of iterations. Since there are no obvious flat components in any of the plots, the sample values seem to take a random walk over their range, so the samples appear to converge to their respective distributions.

Question 5

Now that I have the results from one chain, I'll repeat **Question 3** four more times using different start values of theta, alpha and beta.

```
initial_2 <- c(rep(5, 7), 0.5, 0.5)
initial_3 <- c(rep(1, 7), 10, 10)
```

```
initial_4 <- c(rep(0, 7), 1, 1)
initial_5 <- c(rep(0, 7), 0.1, 0.1)
sample_2 <- gibbs(initial_2, yhat, t, 10000)
sample_3 <- gibbs(initial_3, yhat, t, 10000)
sample_4 <- gibbs(initial_4, yhat, t, 10000)
sample_5 <- gibbs(initial_5, yhat, t, 10000)
```

Question 6

After implementing the Metropolis Hastings-within-Gibbs algorithm five times, I now need to examine the convergence of each parameter over all samples.

```
# Convert samples to MCMC objects
chain_1 <- mcmc(sample_1)
chain_2 <- mcmc(sample_2)
chain_3 <- mcmc(sample_3)
chain_4 <- mcmc(sample_4)
chain_5 <- mcmc(sample_5)

# Store MCMC chain
combined.chains <- mcmc.list(chain_1, chain_2, chain_3, chain_4, chain_5)

# Run Gelman-Rubin diagnostic
gelman.rubin <- gelman.diag(combined.chains)
gelman.rubin

## Potential scale reduction factors:
##
##       Point est. Upper C.I.
##  [1,]       1.00       1.00
##  [2,]       1.00       1.00
##  [3,]       1.00       1.00
##  [4,]       1.00       1.00
##  [5,]       1.00       1.00
##  [6,]       1.00       1.00
##  [7,]       1.00       1.00
##  [8,]       1.01       1.01
##  [9,]       1.00       1.01
##
## Multivariate psrf
##
## 1
```

The Gelman-Rubin diagnostic compares the variability within a single chain to the variability among chains with different initialized values, like the chains computed in **Question 5**. The most important piece of information presented in the table above is the multivariate potential scale reduction feature (psrf), which measures how much the scale of the current distribution would be reduced if it were possible to continue sampling to infinity. A psrf value of 1 indicates the variance within chains and variance across chains are equal. Thus, the Gelman-Rubin diagnostic presents a strong case for convergence in this instance since the multivariate psrf value equals 1.

For each parameter specifically, the maximum upper bound of the psrf is 1.01, which is the case for α and β. The point estimate for α is also 1.01. We generally set a maximum threshold of 1.1 for individual psrf values, so, again, there is a strong case for convergence for each of the 9 parameters.

Question 7

Now that I've assessed the convergence of the parameters over all samples, I report the posterior mean and 95% posterior interval for each in the table shown below.

```
# Combine results from all 5 samples
total.samples <- rbind(sample_1, sample_2, sample_3, sample_4, sample_5)
post.info <- matrix(NA, 9, 3)
for (i in 1:9) {
  post.info[i, 1] <- round(mean(total.samples[,i]), 3)
  post.info[i, 2] <- round(quantile(total.samples[,i], 0.025), 3)
  post.info[i, 3] <- round(quantile(total.samples[,i], 0.975), 3)
}
rownames(post.info) <- c("Theta1", "Theta2", "Theta3", "Theta4",
"Theta5","Theta6","Theta7", "Alpha", "Beta")
colnames(post.info) <- c("Posterior Mean", "Lower Bound", "Upper Bound")
post.info <- as.table(post.info)
names(dimnames(post.info)) <- c("Parameter", "95% Posterior Interval
Information")
post.info
```

```
##          95% Posterior Interval Information
## Parameter Posterior Mean Lower Bound Upper Bound
##    Theta1          5.526       4.385       6.795
##    Theta2          4.874       3.778       6.103
##    Theta3          5.831       4.715       7.083
##    Theta4          5.243       4.173       6.426
##    Theta5          5.327       4.261       6.508
##    Theta6          5.400       4.305       6.600
##    Theta7          5.365       4.328       6.508
##    Alpha           3.474       1.097       7.348
##    Beta            0.633       0.163       1.399
```

Question 8

Hello Small Business Owner,

I'm happy to report that I've been able to find some meaning in the data you've provided me. Given the information on the number of items purchased and length of time a customer spent shopping, I created a statistcal model using the Poisson distribution. The exact details of the distribution are not important here, but it essentially models an expected value in a given period of time. In your case, this would be the expected number of items purchased given a period of time in your store.

I wanted to examine the expected amount of time a customer will take to purchase an item in your store. In order to do this, I used a gamma distribution. Again, the details of the distribution are not important, but there are a few things you should understand:

- α: the number of items a customer will purchase
- β: the mean waiting time until α items are purchased
- θ: the amount of a time it will take a customer to purchase an item in your store

In order to obtain estimate for each of these numbers, I needed to form a distribution around each variable and ensure the distribution accurately represents its respective variable. Fortunately, I was able to confidently model each variable.

Looking at the table I provided you above, there are θ values for each day of the week. Holistically, a customer will take about 5 minutes to purchase an item (looking at the values for θ). The amount of time a customer takes to purchase an item doesn't change much for each day of the week (this is shown by the overlapping intervals in the last two columns), so there's no reason for me to recommend you adjust your sales strategy for different days. However, I would recommend you and your staff focus your time on new customers. If a customer has been in your store for longer than 5 minutes, your efforts are better concentrated on new customers (if present) since there is a greater chance they make a purchase than someone who's been in your store for longer than 5-7 minutes.

Please let me know if there's anything else you'd like me to help you out with.

Best Regards, Jake

Professor Ciminelli,

First and foremost, I would like to apologize for turning this assignment in late. I struggled with sampling from a truncated normal distribution, first trying to use the dtmvnorm function. It wouldn't work because the parameters need to be lists in the vector format c(). Upon adjusting the inputted variables as singular vectors, I tried sampling again and it froze R. I had to force quit R, which led to me losing all of my code and thus restarting the assignment.

I understand I have been a very poor student this semester, and, if I'm punished for turning this assignment in late, I completely understand. After realizing I was not going to make the deadline, I decided to instead focus on the content of this Final Capstone instead of rushing to get it turned in as fast as possible, and I've enjoyed developing a more thorough understanding of the course material because of it.

However, I didn't want to write this message to try and earn more points for the class (I'm well-aware of how poorly I did on the in-class final), but to instead thank you for being an amazing professor. My status as a poor student had nothing to do with your teaching, and I accept all responsibility for my poor performance in your class. I wanted to let you know that you have an amazing impact on all of your students. In the math and statistics departments here at UR, most of the classes are taught by professors who don't care very much about (1) teaching and (2) their students. You are a bright light in the statistics department, and I hope you can change both the culture and content of the curriculum as you progress in your career.

I have spoken to many students about your teaching, and every one of them have spoken very highly of you as a professor and a human being. You have been incredibly understanding whenever I have failed to turn assignments in on time, even when I failed to communicate why that was the case. The past two years of my life have been very difficult, especially this being my senior spring, and I've been forced into many unfortunate circumstances in regards to my academics because of it. Other professors tend to be much less willing to give students second chances, but this is not the case for yourself. I regret not going to class as frequently as I should've; visiting your office hours to enhance my understanding of course material; and being as diligent with course material. Nonetheless, I have thoroughly enjoyed this class and its content, and I hope you maintain your enthusiasm in teaching and finding new courses that can be helpful to students here.

I would like to reiterate that I am not writing this message to receive a better grade on the Final Capstone or in the class as a whole. I understand you probably have many students coming to you at the end of the semester trying to explain their circumstances and help their grade, but this is not the case for myself. I just wanted to let you know the positive impact you have on students in general, even bad students you don't come into contact with as much.

I hope you have a great summer and that all the Final Capstones don't take forever to grade (I've been there).

Regards, Jake